

1 **What is claimed is:**

2

3 1. A computer system employing management software written in a first computer
4 language compatible with first architecture and not compatible with second architecture,
5 said system comprising:

6 a schema formed within said first architecture;

7 header files contained within said schema, said header files being represented in
8 said first language and capable of being utilized by said management software;

9 means for manipulating said header files to locate public functions and/or data
10 attributes of said header files;

11 means, responsive to operation of said manipulating means, for emitting code that
12 calls said public functions and/or data attributes in said first language to obtain called
13 public functions and/or data attributes; and,

14 means for converting said called public functions and/or data attributes to
15 representations of said called public functions and/or data attributes formed in a different
16 computer language compatible with said second architecture.

17

18 2. The computer system of claim 1 further comprising means for forwarding said
19 representations to desired destinations within and beyond said system.

20

21 3. The computer system of claim 1 and wherein said first computer language is
22 RAID++ and said different computer language is XML/CIM.

23

1 4. The computer system of claim 1 and wherein said first computer language is an
2 object-oriented language defining computer data and commands as objects, said
3 manipulating means comprising:

4 means for opening at least one of said header files containing a declaration
5 of at least one of said objects;

6 means for parsing said at least one of said header files to obtain name of
7 class and name of parent class to which said at least one of said objects belongs; and,

8 means for creating a subroutine for accepting said at least one of said
9 objects in said first computer language and generating the equivalent of said at least one
10 of said objects in a different computer language compatible with said second architecture.

11
12 5. The computer system of claim 1 further comprising means for inhibiting initiation
13 of operation of said converting means until said public functions and/or data attributes of
14 said header files are located.

15
16 6. The computer system of claim 1 further comprising means for initiating operation
17 of said converting means upon locating the first of any one of said public functions
18 and/or data attributes.

19
20 7. The computer system of claim 1 and wherein said first computer language is C++
21 and said different computer language is XML/CIM.

22

1 8. The computer system of claim 1 and wherein said first computer language is a
2 first object-oriented language capable of pictorial representation typically in a parent-
3 child tree configuration and said different computer language is a second object-oriented
4 language capable of pictorial representation typically in a flat database configuration.

5
6 9. The computer system of claim 1 further comprising means for inhibiting initiation
7 of operation of said converting means until said public functions and/or data attributes of
8 at least one of said header files are located.

9
10 10. The computer system of claim 1 and wherein said management software is
11 storage management software.

12
13 11. The computer system of claim 1 and wherein said management software is
14 selected from the group consisting of storage, printer, server and other-component
15 management software.

16
17
18 12. A computer network employing a computer system utilizing management
19 software written in a first computer language compatible with first architecture and not
20 compatible with second architecture, said network comprising:

21 a schema formed within said first architecture;

22 header files contained within said schema, said header files being represented in

23 said first language and capable of being utilized by said management software;

1 apparatus for manipulating said header files to locate public functions and/or data
2 attributes of said header files; and,
3 apparatus, responsive to operation of said manipulating apparatus, for emitting
4 code that calls said public functions and/or data attributes in said first language to obtain
5 called public functions and/or data attributes and that converts said called public
6 functions and/or data attributes to representations of said called public functions and/or
7 data attributes formed in a different computer language compatible with said second
8 architecture.

10 13. The network of claim 12 and wherein said computer system further comprises
11 apparatus for forwarding said representations to desired destinations within and outside
12 of said network.

14 14. The network of claim 12 and wherein said first computer language is RAID++
15 and said different computer language is XML/CIM.

17 15. The network of claim 12 and wherein said first computer language is an object-
18 oriented language defining computer data and commands as objects, said manipulating
19 apparatus comprising:

20 apparatus for opening one of said header files containing a declaration of
21 one of said objects;

22 apparatus for parsing said one of said header files to obtain name of class
23 and name of parent class to which said one of said objects belongs; and,

1 apparatus for creating a subroutine for accepting said one of said objects in
2 said first computer language and generating the equivalent of said one of said objects in a
3 different computer language compatible with said second architecture.

4
5 16. The network of claim 12 further comprising apparatus for inhibiting initiation of
6 operation of said converting apparatus until said public functions and/or data attributes of
7 said header files are located.

8
9 17. The network of claim 12 further comprising apparatus for initiating operation of
10 said converting apparatus upon locating the first of any one of said public functions
11 and/or data attributes.

12
13 18. The network of claim 12 and wherein said first computer language is C++ and
14 said different computer language is XML/CIM.

15
16 19. The network of claim 12 and wherein said first computer language is a first
17 object-oriented language capable of pictorial representation typically in a parent-child
18 tree configuration and said different computer language is a second object-oriented
19 language capable of pictorial representation typically in a flat database configuration.

20
21 20. The network of claim 12 further comprising apparatus for inhibiting initiation of
22 operation of said converting apparatus until said public functions and/or and data
23 attributes of at least one of said header files are located.

1

2 21. The network of claim 12 further comprising a SAN which communicates with and
3 is controlled by said computer system.

4

5 22. The network of claim 12 and wherein said management software is storage
6 management software.

7

8 23. The network of claim 12 and wherein said management software is selected from
9 the group consisting of storage, printer, server and other-component management
10 software.

11

12

13 24. A method for utilizing architecture to be practiced in a computer system
14 employing management software written in a first computer language compatible with
15 first architecture and not compatible with said architecture, said method comprising:

16 said management software utilizing a schema having header files in said first
17 language;

18 manipulating said header files to locate public functions and/or data attributes of
19 said header files; and,

20 responsive to operation of said manipulating, emitting code that calls said public
21 functions and/or data attributes in said first language to obtain called public functions
22 and/or data attributes and converts said called public functions and/or data attributes to

1 representations of said called public functions and/or data attributes formed in a different
2 computer language compatible with said architecture.

3
4 25. The method of claim 24 further comprising forwarding said representations to
5 desired destinations within and beyond said system.

6
7 26. The method of claim 25 and wherein said first computer language is RAID++ and
8 said different computer language is XML/CIM.

9
10 27. The method of claim 25 and wherein said first computer language is an object-
11 oriented language defining computer data and commands as objects, said manipulating
12 comprising:

13 opening one of said header files containing a declaration of one of said
14 objects;

15 parsing said one of said header files to obtain name of class and name of
16 parent class to which said one of said objects belongs; and,

17 creating a subroutine for accepting said one of said objects in said first
18 computer language and generating the equivalent of said one of said objects in a different
19 computer language compatible with said architecture.

20
21 28. The method of claim 27 further comprising inhibiting initiation of operation of
22 said converting until said public functions and/or data attributes of said header files are
23 located.

1
2 29. The method of claim 28 further comprising initiating operation of said converting
3 upon locating the first of any one of said public functions and/or data attributes.
4

5 30. The method of claim 29 and wherein said first computer language is C++ and said
6 different computer language is XML/CIM.
7

8 31. The method of claim 27 and wherein said first computer language is a first object-
9 oriented language capable of pictorial representation typically in a parent-child tree
10 configuration and said different computer language is a second object-oriented language
11 capable of pictorial representation typically in a flat database configuration.
12

13 32. The method of claim 24 further comprising inhibiting initiation of operation of
14 said converting until said all public function and data attributes of at least one of said
15 header files are located.
16

17 33. The method of claim 31 further comprising inhibiting initiation of operation of
18 said converting until said public functions and/or and data attributes of at least one of said
19 header files are located.
20

21 34. The method of claim 24 and wherein said architecture is preferred non-legacy
22 architecture.
23

1 35. The method of claim 24 and wherein said management software is storage
2 management software.

3
4 36. The method of claim 24 and wherein said management software is selected from
5 the group consisting of storage, printer, server and other-component management
6 software.

7
8
9 37. A computer program product including management software written in a first
10 language for operation on a computer system designed in accordance with first
11 architecture and not compatible with other architecture, said computer program product
12 comprising:
13 programmable code for utilizing a schema having header files in said first
14 language;
15 programmable code for manipulating said header files to locate public functions
16 and/or data attributes of said header files; and,
17 programmable code, responsive to said manipulating, for emitting special code
18 that calls said public functions and/or data attributes in said first language to obtain called
19 public functions and/or data attributes and converts said called public functions and/or
20 data attributes to representations of said called public functions and/or data attributes
21 formed in a different computer language compatible with said other architecture.

22

1 38. The computer program product of claim 37 further comprising programmable
2 code for forwarding said representations to desired destinations within and beyond said
3 system.

4
5 39. The computer program product of claim 38 and wherein said first computer
6 language is an object-oriented language defining computer data and commands as
7 objects, said programmable code for manipulating comprising:

8 programmable code for opening one of said header files containing a
9 declaration of one of said objects;

10 programmable code for parsing said one of said header files to obtain
11 name of class and name of parent class to which said one of said objects belongs; and,

12 programmable code for creating a subroutine for accepting said one of
13 said objects in said first computer language and generating the equivalent of said one of
14 said objects in a different computer language compatible with said other architecture.

15
16 40. The computer program product of claim 39 and wherein said first computer
17 language is RAID++ and said different computer language is XML/CIM.

18
19 41. The computer program product of claim 40 further comprising programmable
20 code for inhibiting initiation of operation of said programmable code for converting until
21 said public functions and/or data attributes of said header files are located.

22

1 42. The computer program product of claim 40 further comprising programmable
2 code for initiating operation of said programmable code for converting upon locating the
3 first of any one of said public functions and/or data attributes.

4
5 43. The computer program product of claim 39 and wherein said first computer
6 language is C++ and said different computer language is XML/CIM.

7
8 44. The computer program product of claim 37 and wherein said first computer
9 language is a first object-oriented language capable of pictorial representation typically in
10 a parent-child tree configuration and said different computer language is a second object-
11 oriented language capable of pictorial representation typically in a flat database
12 configuration.

13
14 45. The computer program product of claim 44 further comprising means for
15 inhibiting initiation of operation of said converting means until said public functions
16 and/or and data attributes of at least one of said header files are located.

17
18 46. The computer program product of claim 37 and wherein said first architecture is
19 legacy architecture and said other architecture is preferred non-legacy architecture.

20
21 47. The computer program product of claim 37 and wherein said management
22 software is storage management software.

23

1 48. The computer program product of claim 37 and wherein said management
2 software is selected from the group consisting of storage, printer, server and other-
3 component management software.

4

5

6 49. A computer program product compatible with preferred non-legacy architectures
7 and operating in a computer system employing management software written in a first
8 computer language compatible with legacy architecture and not compatible with said
9 preferred non-legacy architecture, said computer program product comprising:

10 programmable code for utilizing a schema having header files in said first
11 language;

12 programmable code for manipulating said header files to locate public functions
13 and/or data attributes of said header files;

14 programmable code, responsive to said manipulating, for emitting special code
15 that calls said public functions and/or data attributes in said first language to obtain called
16 public functions and/or data attributes; and,

17 programmable code for converting said called public functions and/or data
18 attributes to representations of said called public functions and/or data attributes formed
19 in a plurality of different computer languages each being compatible with at least one of
20 said preferred non-legacy architectures.

21

22 50. The computer program product of claim 49 and wherein said management
23 software is storage management software.

1

2 51. The computer program product of claim 49 and wherein said management
3 software is selected from the group consisting of storage, printer, server and other-
4 component management software.

5

6

7 52. In a computer network including a computer system having a functional system
8 therein with management software including a schema for managing said functional
9 system under control of said computer system in accordance with first architecture, a
10 translator-compiler for permitting communication about said managing said functional
11 system to be transmitted between said computer system and other computer devices
12 operating under second architecture, said translator-compiler comprising:

13 program code for accessing header files within said schema to obtain a header file
14 containing particular information;

15 program code for parsing said header file to obtain a particular result;

16 program code for opening an output file for storage of other particular information
17 related to said particular result;

18 program code for continued parsing of said header file to locate public functions
19 and/or data attributes; and,

20 program code for emitting special code to said output file that calls said public
21 functions and/or data attributes to obtain called public functions and/or data attributes and
22 for converting said called public functions and/or data attributes to language compatible
23 with said second architecture;

1 whereby communication about managing said functional system transmitted
2 between said computer system and devices operating under said second architecture is
3 obtained.

4
5 53. In the computer network of claim 52 and wherein said functional system is a
6 storage system and said management software is storage management software.

7
8 54. In the computer network of claim 52 and wherein said functional system is a SAN
9 and said management software is SAN management software.

10
11 55. In the computer network of claim 52 and wherein said first architecture is legacy
12 architecture and said second architecture is non-legacy architecture.

13
14 56. In the computer network of claim 52 and wherein said functional system is
15 selected from the group consisting of storage system, printer system, server system or
16 other-component system and said management software is selected from the group
17 consisting of storage management software, printer management software, server
18 management software and other-component management software respectively.

19
20 57. In the computer network of claim 52 and wherein said other computer devices are
21 located within said network.

1 58. In the computer network of claim 52 and wherein said other computer devices are
2 located outside of said network.

3

4

5 59. In a computer network including a computer system and a functional system
6 controlled by said computer system, management software compatible with legacy
7 architecture having header files, said management software being deployed on both said
8 computer system and said functional system, said management software comprising:

9 translator software means for receiving and manipulating said header files;

10 software means for receiving first requests in first language incompatible with
11 said legacy architecture;

12 software means responsive to operation of said translator software means for
13 obtaining responses to said first requests in second language compatible with said legacy
14 architecture; and,

15 software means for converting said responses to equivalent responses compatible
16 with said first language and for communicating said equivalent responses to the
17 destination from which, or to destinations related to that from which, said first requests
18 originated.

19

20 60. In the computer network of claim 59 and wherein said functional system is a
21 storage system.

22

1 61. In the computer network of claim 59 and wherein said functional system is a
2 SAN.

3
4 62. In the computer network of claim 59 and wherein said first requests are received
5 from outside of said network.

6
7
8 63. In a computer system compatible with computer architecture, management
9 software comprising:
10 software means for receiving first requests in first language incompatible with
11 said computer architecture;
12 software means for obtaining responses to said first requests in second language
13 compatible with said computer architecture; and,
14 software means for converting said responses to equivalent responses compatible
15 with said first language and for communicating said equivalent responses to the
16 destination from which, or to destinations related to that from which, said first requests
17 originated.

18
19 64. In the computer system of claim 63 and wherein said computer architecture is
20 legacy architecture.

21
22 65. In the computer system of claim 64 and wherein said management software is
23 storage management software.

1

2 66. In the computer system of claim 64 and wherein said management software is
3 SAN management software.

4

5 67. In the computer system of claim 66 and wherein said destination is located
6 outside of said computer system.

7

8

9 68. A computer program product to be operated on a computer compatible with
10 computer architecture comprising:

11 programmable code for receiving first requests in first language incompatible with
12 said computer architecture;

13 programmable code for obtaining responses to said first requests in second
14 language compatible with said computer architecture; and,

15 programmable code for converting said responses to equivalent responses
16 compatible with said first language and for communicating said equivalent responses to
17 the destination from which, or to destinations related to that from which, said first
18 requests originated.

19

20 69. The computer program product of claim 68 and wherein said computer
21 architecture is legacy architecture.

22

23

1 70. A method for managing functional systems to be practiced on a computer
2 compatible with computer architecture comprising:
3 receiving first requests in first language incompatible with said computer
4 architecture;
5 obtaining responses to said first requests in second language compatible with said
6 computer architecture; and,
7 converting said responses to equivalent responses compatible with said first
8 language and communicating said equivalent responses to the destination from which, or
9 to destinations related to that from which, said first requests originated.

10
11 71. The method of claim 70 and wherein said computer architecture is legacy
12 architecture.

13
14 72. The method of claim 71 and wherein said functional systems include a storage
15 system.

16
17 73. The method of claim 71 and wherein said functional systems include a SAN.

18
19 74. The method of claim 71 and wherein said functional systems are selected from the
20 group consisting of storage systems, printer systems, server systems, and other-
21 component systems.

75. In a computer network including a computer system and a storage system controlled by said computer system, a method for managing storage compatible with architecture having header files, said method being deployed on both said computer system and said storage system, said method comprising:

- translating and manipulating said header files to obtain translated and manipulated header files;
- receiving first requests from outside of said network in first language incompatible with said architecture;
- in cooperation with said translated and manipulated header files, obtaining responses to said first requests in second language compatible with said architecture; and,
- in cooperation with said translated and manipulated header files, converting said responses to equivalent responses compatible with said first language and communicating said equivalent responses to said outside of said network.

76. The method of claim 75 and wherein said architecture is legacy architecture.

77. In the computer network of claim 76 further comprising said storage system is a SAN.

78. In the computer network of claim 75 and wherein said first language is a first object-oriented language capable of pictorial representation typically in a flat database configuration and said second language is a second object-oriented language capable of pictorial representation typically in a parent-child tree configuration.

1

2 79. In the computer network of claim 78 and wherein said first language is CIM/XML
3 and said second language is C++.

4

5 80. In the computer network of claim 79 and wherein said C++ language is RAID++.

6

7

8 81. In an improved network including a first computer network operating in
9 accordance with first architecture and a second computer network operating in
10 accordance with second architecture, the improvement comprising:

11 an interface between said first computer network and said second computer
12 network to automatically convert communication from said second computer network
13 into a form compatible with said first computer network, and to automatically convert
14 response to said communication generated by said first computer network into a form
15 compatible with said second computer network.

16

17 82. In the improved network of claim 81 and wherein said first architecture is legacy
18 architecture and said second architecture is non-legacy architecture.

19

20 83. The improvement of claim 82 and wherein said first computer network operates in
21 accordance with said legacy architecture supporting a first object-oriented computer
22 language capable of pictorial representation typically in a parent-child tree configuration,
23 and wherein said second computer network operates in accordance with said non-legacy

1 architecture supporting a second object-oriented computer language capable of pictorial
2 representation typically in a flat database configuration.

3
4 84. The improvement of claim 83 and wherein said first object-oriented computer
5 language is C++ and wherein said second object-oriented computer language is
6 XML/CIM.

7
8 85. The improvement of claim 84 and wherein said communication includes
9 management software communication.

10
11 86. The improvement of claim 85 and wherein said management software
12 communication includes storage management software communication.

13
14 87. The improvement of claim 86 and wherein said storage management software
15 communication relates to SAN communication.

16
17 88. The improvement of claim 85 and wherein said management software
18 communication includes storage, printer, server, and other-component management
19 software communications.

20
21 89. The improvement of claim 81 and wherein said response is communicated to the
22 destination from which, or to destinations related to that from which, said communication
23 originated.